DOCUMENT RESUME

ED 385 285 IR 055 569

AUTHOR Schmaranz, Klaus
TITLE The Hyper-G Information System.
PUB DATE 31 May 95
NOTE 9p.
PUB TYPE Reports - Descriptive (141)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Computer Networks; *Computer Software Development; Databases; Foreign Countries; *Hypermedia; *Information Systems; *Online Searching; Technological Advancement
IDENTIFIERS Boolean Search Strategy; Linkage

ABSTRACT

This paper provides a brief description of Hyper-G, the first second-generation hypermedia system. The first section identifies problems with first generation hypermedia systems. The following sections discuss the new concepts that are implemented in Hyper-G, including user accounts and billing, structuring of data, caching and replication, native document types, navigation, and editing and authoring. These concepts, such as a world-wide distributed network database, a separated link database and bidirectional links allow for highly sophisticated navigation and hyperlinks in all native document types such as hypertext, images, Postscript documents and even in movies, sound, and 30 scenes. Hyper-G also implements powerful search mechanisms, such as Boolean searching of titles, keywords, and fulltext with user-defined scope from one collection on one server to all servers worldwide. The system is compatible to first generation systems like Gopher and World Wide Web. (Author/AEF)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED 385 285

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
☐ This document has been reproduced as received from the person or organization originating it.
☐ Minor changes have been made to improve reproduction quality.
• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

The Hyper-G Information System

Klaus Schmaranz
(Institute for Information Processing and
Computer Supported New Media (IICM),
Graz University of Technology, Austria
kschmar@iicm.tu-graz.ac.at)

May 31, 1995

Abstract

In this paper a brief description of Hyper-G - the first second generation hypermedia system - is given. The first section identifies some problems with first generation hypermedia systems. The following sections are an outline of the new concepts that are implemented in Hyper-G. These new concepts such as a world-wide distributed network database, a separated link database and bidirectional links allow highly sophisticated navigation, hyperlinks in all native document types such as hypertext, images, PostScript documents and even in movies, sound and 3D scenes. Hyper-G also implements very powerful search mechanisms such as boolean search on titles, keywords, even fulltext with user defined scope from one collection on one server to all servers worldwide. And last but not least Hyper-G is fully compatible to first generation systems like Gopher and WWW.

1 Introduction

Currently, the most p ular Internet information systems use distributed menus and searching (Gopher, see [Alberti et al 1992]) or hypertext documents (WWW, see [Berners-Lee et al 1994]) to represent information spaces.

In both cases information is stored in a simple structured fashion on servers, and can be accessed via clients, with clients available for most major hardware platforms. Both systems provide an easy to use interface and are easy to install for service providers as well as for readers, but both systems also have some well-known weaknesses:

IR055569

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Di Klaus Schmaranz
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

BEST COPY AVAILABLE

- In both systems the knowledge of certain server addresses is required for browsing.
- In WWW Hyperlinks as the only structuring tool (Robert Cailliau: "the world wide spaghetti bowl") lead to the "lost in hyperspace" syndrom: after following a number of links it is nearly impossible to find the way to an interesting page again the next time a reader searches it. Furthermore there is no way to decide how much of the information has already been seen because the hierarchical structure is missing.
- Hyperlinks are made up using URLs (Universal Resource Locators) - they point to the location of a document. If a document moves or disappears the hyperlinks point to nowhere. It is desirable to make up hyperlinks using URNs (Universal Resource Names) that identify the document itself instead of the location.
- Both Gopher and WWW servers are difficult to administrate as the number of documents grows because there is no underlying database.
- Search mechanisms, depending on the search engine, are mostly limited to a simple Title search.
- Both Gopher and WWW are lacking user identification and billing mechanisms.

Hyper-G [see Maurer 1994], [Andrews et al 1994] as the first second generation hypermedia system tries to overcome the weaknesses mentioned above using a completely new underlying concept [see Kappe et al 1993]. One of the major design goals of Hyper-G was it to be fully compatible to Gopher and WWW and therefore to fit perfectly into the existing Internet landscape. Naturally the modern design of Hyper-G allows to add many new sophisticated navigation, editing and authoring features as you will read in the following sections. With some lack of the special Hyper-G functionality Hyper-G servers can naturally be accessed with all well-known Gopher and WWW clients.

2 The Hyper-G Database

Hyper-G is based on a client-server architecture with servers available for all major Unix platforms and clients available for Unix and MS-Windows platforms. A native Macintosh client is under development. Naturally one can also use all well-known Gopher and WWW clients to access a Hyper-G server.

The client-server protocol of Hyper-G is a connection oriented protocol instead of a connectionless protocol as implemented in Gopher or WWW. This allows it to implement user access rights and billing mechanisms in Hyper-G.

The Hyper-G server itself is based on a distributed network database architecture with separated database engines for documents and hyperlinks. Using this concept the whole world of Hyper-G servers presents itself to the user as one large single database with no need to know server addresses to access single servers. The server structure is completely hidden to the user.

Each document in Hyper-G has an additional Object description where all the important Attributes of a document are stored. Attributes include title, keywords, type, creation time, last modification time and also some very special attributes like the expiry date of a document if any. Expiry dates are very important for quality assurance - who has not already seen a call for papers for a conference that was months ago! In Hyper-G a document is no longer accessible after the expiry date and also all the links pointing to it are invisible.

Object attributes are searchable allowing very sophisticated searches like "find all documents that are written by a special author and not older than half a year" or "find all text or PostScript documents that have the term multimedia in their title or keywords".

All Objects in Hyper-G have a world-wide unique Object ID that allows to locate them even across server boundaries.

Due to the separate link database all links in Hyper-G are bidirectional. Visiting a document a reader can not only keep track of all the links pointing from this document to the outside world but also can have a look at all links pointing from outside to this document.

The second and much more important feature of bidirectional links is that they allow to keep the link structure consistent! If a document is moved from one location to another, even across server boundaries the link is pointing to the document at its new location. Hyperlinks in Hyper-G are based on the concept of URNs (Universal Resource Names) instead of URLs (Universal Resource Locators).

Another very important feature of the separated link database is the possibility to add links to all supported document types, even hyperlinks to and from images, PostScript documents, movies, 3D scenes and sound are supported! Since the links are stored outside the document there is no need to change format specifications of standard formats to make them hyperlinkable.

3 User Accounts and Billing

Handling user accounts in Hyper-G is done similar to the way Unix handles them: each user can belong to one or more groups and each user has a personal home-collection where he can store his personal documents or references to existing documents. Access to documents is then controlled by read, write, and unlink permissions.

Billing mechanisms in Hyper-G are at the moment implemented following 2 different paradigms: The first method is to give the user a certain amount of "money" and to give charged documents a certain price. Each time a charged document is downloaded the price of the document is subtracted from the actual funds of the user. In this case the user pays in advance.

The second method is based on the library paradigm and mostly used for electronic publishing: The library that operates the server subscribes for a certain number of exemplars of a journal, e.g. 3 exemplars. Users have access to the journals on the library server, depending on their access rights, in the case of libraries normally also anonymous users have access. But for e.g. 3 exemplars one document is only accessible by a maximum of 3 simultaneous users, and after a user has "borrowed" the document it is not accessible for a given time, e.g. 1 hour. If more than the maximum allowed number of users wants to get access the users get a "at the moment not available" message.

4 Structuring of data

Structuring of documents in Hyper-G is done using a collection hierarchy very similar to Gopher's file hierarchy. Additional structuring tools are Hyperlinks as known from WWW and a special clustering mechanism. Documents can be members of arbitrary many collections, even across server boundaries, in this case they are represented as references in the collection tree. Collections can naturally not only hold native Hyper-G objects but also references to Gopher and WWW documents, telnet objects and many more. Also completely exotic document types can be referenced by a special generic object that holds the object description and additional information, e.g. which viewer to start for this type of object.

Clusters in Hyper-G are several documents that are grouped together for logical reasons. One example of a logical cluster would be an image and a text describing it or a movie and additional sound to the movie. When clicking on a cluster the reader gets all documents contained, in case of the movie and sound example the movie player will be started and the sound played simultaneously.

One even more important feature of clustering is the support of multilinguality. One can store several documents in different languages in a cluster, even togegther with other arbitrary documents. Hyper-G users have the possibility to choose their favourite language and in case of a multilingual cluster will get the document in their preferred language if it exists. One could for example store an image together with an English and a German description in a clusters and users that have chosen English as their preferred language would then get the image with the English description.

5 Caching and Replication

Due to the exponential growth of Internet and of information stored in Internet scalability of Hyper-G was an important design goal. During rush hours data transfer over the Internet is unacceptably slow, especially for overseas connections. Due to this fact caching and mirroring mechanisms had to be implemented.

Caching in Hyper-G is implemented using well-known principles, but there is another specialized mechanism used for mirroring data between servers - replication:

Documents already existing on one server can be exported from the database and mirrored to other servers. In this case they should obtain a special replica-attribute to let the server know that they are mirrored objects from another server. The server then evaluates the replica-attribute for remote objects as follows:

Let us assume following scenario: A user in the USA starts his Hyper-G client and connects to his nearest Hyper-G server. Since the Hyper-G database seamlessly hides server boundaries this user surfs a little bit in hyperspace until he finds an interesting document in Austria. He decides to get this document. Normally all data would be transferred from Austria to the USA. But this document is for example a paper of an electronic journal and the journal is completely mirrored to the server in the USA - only the user did not know. In this case the Hyper-G server in the USA automatically detects that it has stored a replica of the document of Austria and sends the user the local replica instead of unnecessarily downloading the document from Austria.

6 Native Document Types

Hyper-G supports a number of native document types as well as external documents such as references to Gopher and WWW documents and generic objects.

Native document types are at the moment:

- Images of many different formats

- MPEG movies
- PostScript documents
- Sound files
- 3D scenes.

All of the above formats are fully hyperlinkable - in the case of MPEG movies the clickable areas of hyperlinks even can move and change their size and shape in a hyperlinked film.

PostScript documents are often used for archive material that is not in hypertext format or also very often for specialized text, e.g. mathematical papers containing lots of formulae that cannot be handled very well by simple hypertext formats. Full support of hyperlinks in PostScript adds the functionality that one normally knows from hypertext. To add full hypertext functionality a method to do full text search in PostScript files is under development and will be available soon. Also an advanced PDF viewer similar to Adobe Acrobat is under development.

But, Hyperlinkable does not only mean that one can define clickable areas in documents, so called source anchors. One can define destination anchors such as a special sequence in a soundfile. One could for example have a text document with the text of a song and if one clicks on a certain text line the sound player jumps to that part and plays the music that belongs to this line.

Of special interest especially for the future when 3D hardware gets cheaper and computing power increases even for the low end computers is the 3D support. Hyper-G at the moment supports 3D scenes in WaveFront format naturally again with full hyperlink support.

Together with the University of Minnesota and NCSA full VRML support is under development, the first prototype of this new 3D viewer generation is already implemented in Harmony - the Unix Hyper-G client.

7 Navigation

As has already been said Hyper-G presents itself to the user as a single structured network database. The representation of this network database in Hyper-G clients is normally a directory-tree like structure of collections. In Harmony, the Unix Hyper-G client, also a second representation of the collection hierarchy is implemented - the representation as a 3D information landscape. Collections and documents are represented as buildings in a landscape and the user can fly through the landscape.

Navigation also includes very advanced features such as the local map. Based on the bidirectional nature of links one can create a local map on the fly showing arbitrary levels of incoming and outgoing links. This is very interesting if one finds an interesting document; in most cases other interesting documents on the same topic can be found by following the links to and from this document.

Very important for navigation in Hyperspace is searching. Hyper-G supports boolean title-, keyword- and fulltext-search with user definable scope from one single collection on one single server to all collections on all servers worldwide. But what if the search finds several hundreds of documents? For this case there is a special feature in the clients, the so called "location feedback" - a simple click on a document in the search result window opens the path to the document. So the user can decide if this document is of interest for him. Let's take for example a simple search for the item "grep". The user wanted to know the exact syntax and therefore searches for a manpage on that topic. The search could result in 2 search hits, one in the hacker's jargon and the other one in the manpages. A single click on the first object immediately shows that this hit is in the collection "hacker's jargon" and is therefore not interesting for the user without forcing the user to download the document first.

8 Editing and Authoring with Hyper-G

As already mentioned Hyper-G is not a read-only system where users can easily view documents, Hyper-G databases allow read and write access according to the user access rights. Documents can easily be copied or moved from one collection to another, even across server boundaries and naturally documents can also be edited. Hyperlinks in documents are created very simply using the mouse.

To control the presentation of collections in Hyper-G the author can add Attributes like presentation hints, sort order, sequence number and some more to documents.

Also available for Hyper-G are tools dealing with mass data such as inserting, deleting, changing or replacing huge numbers of documents in a single batch job as well as tools for mirroring data on other servers.

9 Conclusion

Hyper-G is designed to manage huge amounts of information and provide mechanisms making it easy for the information provider to administrate a server as well as it is designed to make the Internet surfer's life easy. Integrated sophisticated search mechanisms as well as new navigation facilities such as a 3D landscape view of the

database strongly ease data retrieval. Hyper-G is a first implementation of the second generation of hypermedia systems whose features nobody will want to miss in the future.

References

[Alberti et al 1992] Alberti,B., Anklesaria,F., Lindner,P., McCahill,M., Torrey,D.; Internet Gopher Protocol: A Distributed Document Search and Retrieval Protocol; FTP from boombox.micro.umn.edu, in directory pub/gopher/gopher_protocol.

[Andrews et al 1994] Andrews,K., Kappe,F.; Soaring Through Hyperspace: A Snapshot of Hyper-G and its Harmony Client; Proc. of Eurographics Symposium on Multimedia/Hypermedia in Open Distributed Environments, Graz (1994).

[Berners-Lee et al 1994] Berners-Lee,T., Cailliau,R., Luotonen,A., Nielsen,H., Secret,A.; The World-Wide Web; Communications of the ACM 37,8 (1994), 76-82.

[Kappe et al 1993] Kappe,F., Maurer,H., Scherbakov,N.; Hyper-G – A Universal Hypermedia System; Journal of Educational Multimedia and Hypermedia 2,1 (1993), 39-66.

[Maurer 1994] Maurer,H.: Advancing the ideas of World-Wide Web; Proc. Conf. on Open Hypermedia Systems, Honolulu (1994), 201-203.